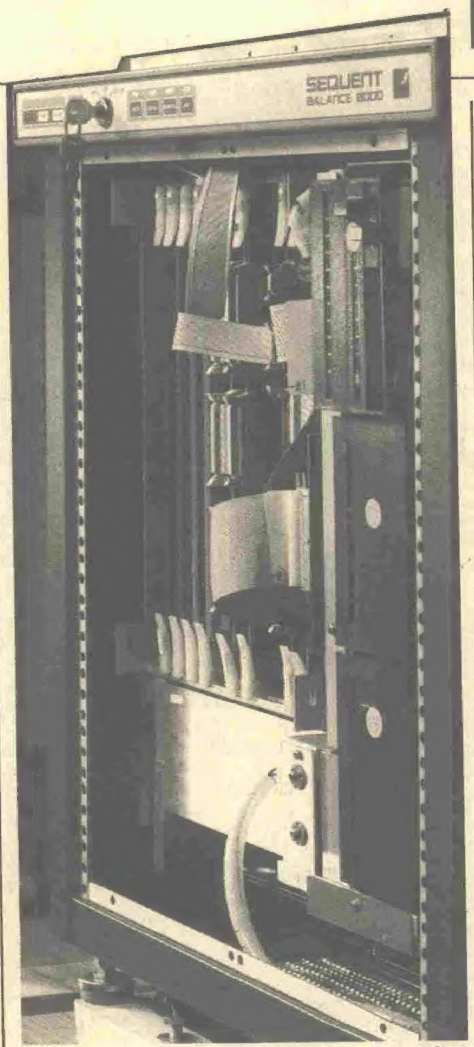
การเขียนโปรแกรมสำหรับการคำนวณแบบขนาน

ตอนที่ 1
แนะนำเครื่องคอมพิวเตอร์แบบหลายโปรเซสเซอร์

จินต์-นุดี วรมหาภูมิ

ในช่วงทศวรรษที่ผ่านมานี้ ความต้องการทางด้านคอมพิวเตอร์ได้เพิ่มขึ้นอย่างรวดเร็ว บรรดานักค้นคว้าและวิจัยพยายามค้นหาวิธีการเพื่อให้คอมพิวเตอร์ทำงานได้ว่องไวและมีประสิทธิภาพเพิ่มขึ้นจากที่มีอยู่ เนื่องจากเทคโนโลยีด้านอิเล็กทรอนิกส์ที่ใช้อยู่ในปัจจุบันเกือบถึงสภาพอิ่มตัว การที่เราจะเพิ่มความเร็วในการคำนวณของเครื่องคอมพิวเตอร์จึงเป็นไปได้ยาก หากจะสร้างคอมพิวเตอร์ที่สามารถคำนวณได้เร็วขึ้นเป็นสองเท่า อาจจะทำให้ราคาของเครื่องแพงมากขึ้นเป็นสิบเท่า เพราะการออกแบบและสร้างเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงต้องใช้เทคโนโลยีขั้นสูง ซึ่งมีราคาแพงและคอมพิวเตอร์ที่มีประสิทธิภาพสูงเหล่านี้ยังมีผู้ซื้อจำกัด ต้นทุนการผลิตวิจัยจึงเป็นสาเหตุให้ราคาเครื่องแพงมากเกินความจำเป็น ตัวอย่างเช่น เครื่องเมนเฟรมชนิดโปรเซสเซอร์เดี่ยวที่ดีที่สุดในปัจจุบัน มีความสามารถในการคำนวณเร็วกว่าเครื่องไมโครคอมพิวเตอร์ทั่ว ๆ ไปประมาณ 50-60 เท่า แต่เมื่อเปรียบเทียบด้านราคาแล้ว เมนเฟรมที่ว่าจะแพงกว่าไมโครประมาณ 1,000 เท่าขึ้นไป

จากการเปรียบเทียบอัตราส่วนระหว่างราคาต่อประสิทธิภาพ (price-performance) ของเครื่องคอมพิวเตอร์แบบเมนเฟรมและไมโครฯแล้ว จะเห็นเห็นได้ว่าเครื่องคอมพิวเตอร์แบบไมโครฯ จะมีอัตราส่วนระหว่างราคาต่อประสิทธิภาพดีกว่าเมนเฟรมมาก ในเมื่อเห็นว่า อัตราส่วนระหว่างราคาต่อประสิทธิภาพของเครื่องคอมพิวเตอร์แบบไมโครฯ ดีกว่าเมนเฟรม หลายคนคงจะสงสัยว่า ทำไมเราจึงไม่ใช้เครื่องคอมพิวเตอร์แบบไมโครฯ สัก 100 หรือ 200 เครื่องมาทำงานแทนเครื่องเมนเฟรมที่มีอยู่ เพื่อช่วยให้เราทำงานได้เร็วขึ้นในราคาที่ถูกลง



เรื่องนี้ฟังดูเหมือนง่าย แต่จริง ๆ แล้วไม่ง่ายเหมือนที่คิด (ไม่เช่นนั้นบริษัทผลิตคอมพิวเตอร์แบบเมนเฟรมคงล้มละลายไปตาม ๆ กันแล้ว) เพราะปัญหาบางประเภท โดยเฉพาะการคำนวณทางคณิตศาสตร์และวิทยาศาสตร์ที่สลับซับซ้อน ถึงจะมีคอมพิวเตอร์หลาย ๆ เครื่องก็ไม่ต่างไปจากมีเครื่องคอมพิวเตอร์เพียงเครื่องเดียวเลย เพราะเราไม่สามารถแบ่งงานการคำนวณออกเป็นชิ้นย่อย ๆ ที่ไม่ขึ้นต่อกันให้คอมพิวเตอร์แต่ละเครื่องคำนวณได้ ตัวอย่างเช่น การแก้ปัญหาสมการเชิงเส้น หรือการหาค่า inverse ของ matrix ขนาดใหญ่ การคำนวณเหล่านี้ บางครั้งอาจกินเวลาเป็นวันหรือเป็นสัปดาห์ในการคำนวณบนเครื่องเมนเฟรมขนาดใหญ่ ถ้าเรานำมาคำนวณบนเครื่องไมโครคอมพิวเตอร์แล้ว อาจจะกินเวลาเป็นปีหรือมากกว่านั้น นอกจากนี้ ระบบคอมพิวเตอร์ที่ต้องการเครื่องเมนเฟรมขนาดใหญ่ ส่วนใหญ่จะเป็นระบบศูนย์กลาง (Centralized) ข้อมูลระบบจะถูกเก็บอยู่ในฐานข้อมูลขนาดใหญ่ ซึ่งมีผู้ใช้ประมาณ 100 หรือ 1,000 ขึ้นไป ผู้ใช้เหล่านี้จะอ่านและแก้ไขข้อมูลผ่านทางสายโทรศัพท์อยู่ตลอดเวลา ข้อมูลภายในฐานข้อมูลจึงจำเป็นต้องถูกปรับปรุงให้ทันสมัยอยู่ตลอดเวลา ไม่เช่นนั้นอาจจะก่อความเสียหายให้แก่องค์การนั้น ๆ อย่างมากมายจนคาดไม่ถึง ตัวอย่างของระบบคอมพิวเตอร์ที่กล่าวถึงนี้ได้แก่ ระบบสำรองที่นั่งของสายการบิน และระบบเงินฝากของธนาคาร เป็นต้น

ในกรณีของระบบศูนย์กลางเช่นนี้ แทบจะเป็นไปไม่ได้ที่เราจะนำไมโคร

คอมพิวเตอร์หลายเครื่องมาทำงานแทนเมนเฟรมคอมพิวเตอร์ในการควบคุมและใช้งานฐานข้อมูลกลาง เพราะถ้าเราปล่อยให้คอมพิวเตอร์หลาย ๆ เครื่องแก้ไขข้อมูลในฐานข้อมูลกลางในเวลาเดียวกันโดยไม่มีการควบคุมที่รัดกุมเพียงพอ ผลลัพธ์ที่ได้อาจจะเป็นข้อมูลที่ผิดพลาด

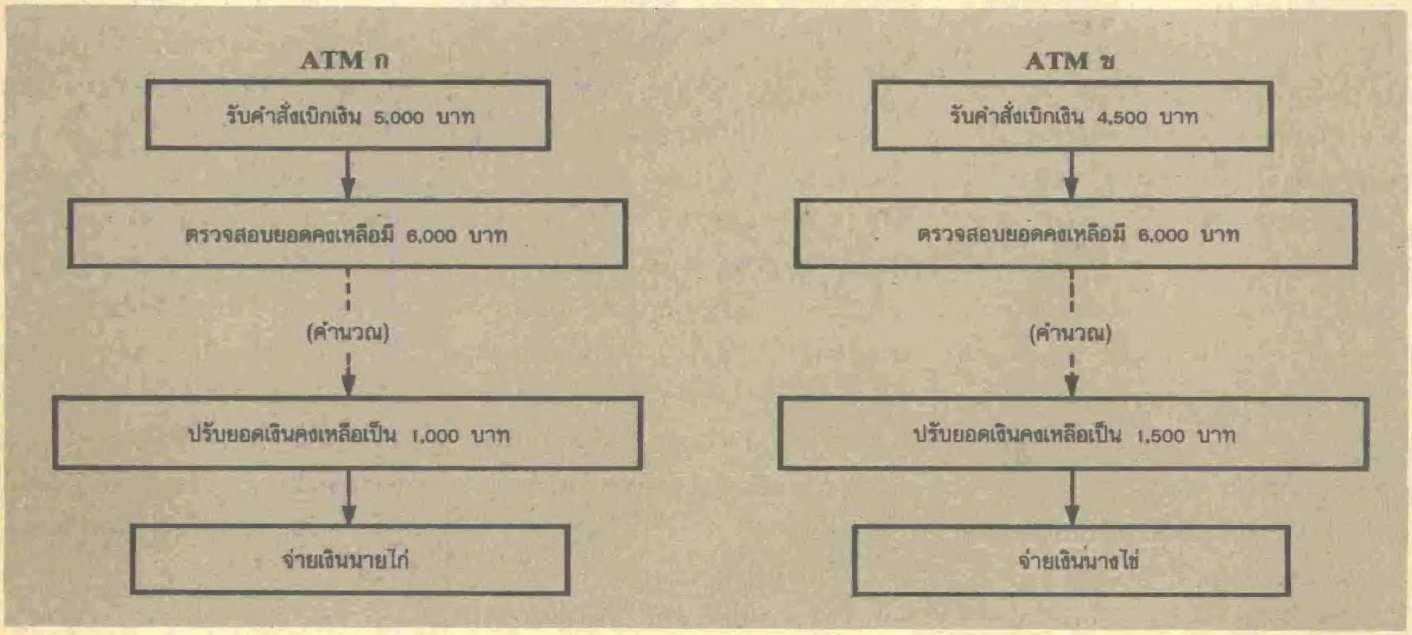
เพื่อให้ผู้อ่านมองเห็นปัญหาได้ชัดเจนยิ่งขึ้น จะขอยกตัวอย่างง่าย ๆ เกี่ยวกับปัญหาการแก้ไขข้อมูลในฐานข้อมูลกลางจากเครื่องคอมพิวเตอร์หลาย ๆ เครื่อง โดยที่ไม่มีเครื่องกลางคอยควบคุมอยู่

สมมติให้ นายไก่ และ นางไข่ สองสามีภรรยา ไปเปิดบัญชีกับธนาคาร "บรรลัย" เป็นจำนวน 6,000 บาท ธนาคารได้ออกบัตร ATM ให้กับสามีภรรยาคู่นี้คนละใบ เพื่อเป็นการเอาใจลูกค้า และแต่ละคนจะได้เบิกเงินหรือฝากเงินได้สะดวกขึ้น เนื่องจากฝ่ายบริหารของธนาคารอยากจะประหยัดเงินสำหรับระบบคอมพิวเตอร์ จึงใช้เครื่องไมโครคอมพิวเตอร์หลายสิบเครื่องที่สำนักงานใหญ่ ทำหน้าที่เป็นตัวกลางระหว่างเครื่อง ATM กับฐานข้อมูลกลาง แทนที่จะใช้เมนเฟรมขนาดใหญ่เพียงเครื่องเดียว ซึ่งมีราคาแพงกว่ามาก อยู่มาวันหนึ่ง ทั้งนายไก่ และนางไข่ จำเป็นต้องใช้เงินขึ้นมาพร้อมกัน ต่างคนต่างก็ใช้การ์ด ATM ของตัวเองไปเบิกเงินกับเครื่อง ATM ที่อยู่ข้างที่ทำงานของแต่ละคน นายไก่ เบิกเงิน 5,000 บาท ในขณะที่นางไข่ เบิกเงิน 4,500 บาท ทั้งคู่สามารถเบิกเงินได้ตามต้องการได้เงินรวมกัน 9,500 บาท ทั้ง ๆ ที่ในบัญชีร่วมของคนทั้งสองมีเพียง 6,000 บาท ความผิดพลาดที่เกิดขึ้นนี้ต้นเหตุไม่ได้จากโปรแกรมเมอร์เขียนโปรแกรมผิด หรือ คอมพิวเตอร์คำนวณผิดพลาด แต่เกิดขึ้นจากการเปลี่ยนแปลงแก้ไขข้อมูลกลาง สามารถทำได้จากหลายแห่งโดยไม่มีการควบคุม ทำให้การทำงานของเครื่องคอมพิวเตอร์แต่ละตัวแทรกแซงซึ่งกันและกัน จากตัวอย่างการเบิกเงินข้างต้นสามารถเขียนเป็นการทำงานตามขั้นตอนได้ดังรูปที่ 1

เราจะเห็นได้ว่า เมื่อเครื่อง ATM ก และ ข ตรวจสอบบัญชีร่วมในเวลาใกล้เคียงกัน เครื่องทั้งสองจะพบว่ามีเงินคงเหลือในบัญชีอยู่ 6,000 บาท ซึ่งมากกว่าจำนวนที่ต้องการเบิก เครื่อง ATM ทั้งสองจึงปรับยอดเงินคงเหลือใหม่ ถ้าสมมติให้เครื่อง ATM ข ทำช้ากว่าเสี้ยววินาทีดังนั้น เครื่อง ATM ข จะแก้ไขยอดเงินคงเหลือทับค่าที่ ATM ก เพิ่งจะเขียนลงไป ทำให้ยอดเงินคงเหลือของบัญชีเป็น 1,500 บาท หลังจากปรับยอดเงินเสร็จ เครื่อง ATM ทั้งสองก็จ่ายเงินให้นายไก่และนางไข่ เป็นเงินรวมกัน 9,500 บาท

ปัญหาที่ยกขึ้นมานี้เกิดขึ้นเพราะเรามีข้อมูลที่จำเป็นจะต้องใช้ร่วมกันโดยโปรแกรมหลาย ๆ โปรแกรม ซึ่งคำนวณอยู่บนเครื่องคอมพิวเตอร์หลาย ๆ เครื่องในเวลาเดียวกัน ผู้เขียนจะกลับมาบรรยายเกี่ยวกับปัญหาที่กล่าวมาข้างต้นนี้อย่างละเอียด พร้อมทั้งวิธีแก้ปัญหาอีกครั้งหนึ่งในฉบับต่อไป เนื่องจากปัญหานี้เป็นปัญหาที่สำคัญที่สุดอันหนึ่งในการออกแบบโปรแกรมสำหรับการคำนวณแบบขนาน

อนึ่ง ปัญหานี้ในระบบคอมพิวเตอร์ทั่วไป จะแก้โดยวิธีการของ operating



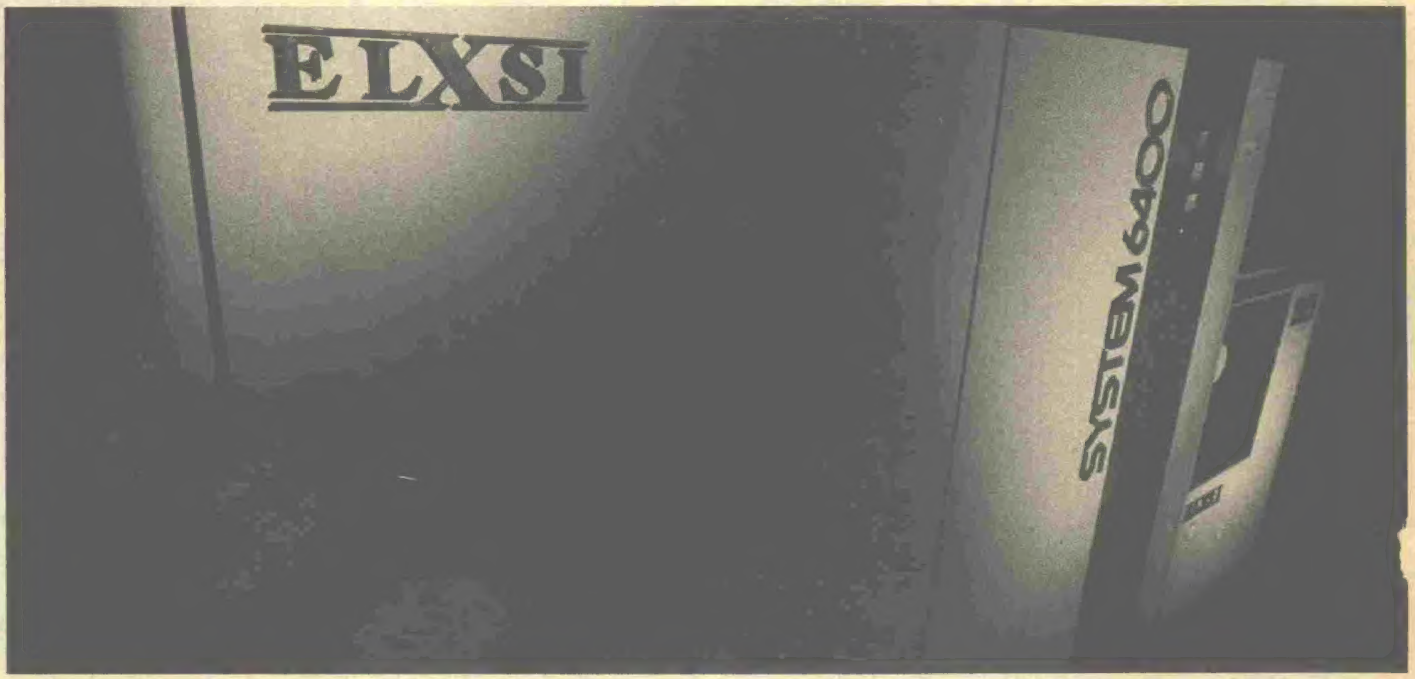
รูปที่ 1 ลำดับการทำงานของเครื่อง

systems เช่น กลไกพวก lock, semaphore เป็นต้น แต่ระบบคอมพิวเตอร์แบบขนาน ปัญหานี้ก็จะพบอีกวิธีการแก้ก็ต้องใช้กลไกคล้ายกัน แต่ไม่เหมือนกันทีเดียวนัก

ในช่วงหลายปีที่ผ่านมา ความสามารถของคอมพิวเตอร์ระดับไมโครฯ ได้รับการพัฒนาขึ้นอย่างรวดเร็ว เพราะความต้องการของตลาดและการแข่งขันที่สูงขึ้นมาก ทำให้เครื่องไมโครคอมพิวเตอร์เหล่านี้มีราคาถูกลงเป็นลำดับ นอกจากนี้ในปัจจุบัน ขนาดของไมโครโปรเซสเซอร์ซึ่งเป็นหัวใจของไมโครคอมพิวเตอร์ยังมีขนาดเล็กลงจนสามารถบรรจุอยู่บนชิพขนาดประมาณหนึ่งตารางเซนติเมตรได้ เนื่องจากขนาดที่เล็กลงและราคาที่ถูกขึ้นของไมโครโปรเซสเซอร์ในปัจจุบัน นักค้นคว้าและวิจัยจึงหันมาสนใจที่จะนำไมโครโปรเซสเซอร์ขนาดเล็กเหล่านี้มาต่อเข้าด้วยกันเป็นระบบหลายโปรเซสเซอร์ (multiprocessor) ที่มีประสิทธิภาพสูงมาก อาจจะสูงกว่าเครื่องซุปเปอร์คอมพิวเตอร์ที่ดีที่สุดในปัจจุบัน จำนวนโปรเซสเซอร์ที่นำมาต่อเข้าด้วยกันนี้มีจำนวนตั้งแต่ สอง สาม สิบ ร้อย พัน หรือเป็นหมื่นโปรเซสเซอร์ ขึ้นอยู่กับเทคนิคที่ใช้และประสิทธิภาพที่ต้องการ

เมื่อแนวโน้มของระบบคอมพิวเตอร์ขนาดใหญ่จะเปลี่ยนไปเป็นระบบหลายโปรเซสเซอร์ ซึ่งประกอบไปด้วยไมโครโปรเซสเซอร์จำนวนมาก ปัญหาที่ตามมาก็คือ จะเขียนโปรแกรมอย่างไรให้สามารถคำนวณได้อย่างมีประสิทธิภาพบนเครื่องคอมพิวเตอร์แบบใหม่นี้ เพราะโดยทั่วไป เราไม่สามารถจะทำอะไรสองอย่างพร้อมกันให้ได้ผลออกมาดีเท่ากัน ลองดูง่าย ๆ ก็ได้เช่น พยายามอ่านหนังสือสองเล่มพร้อม ๆ กัน โดยอ่านหนึ่งบรรทัดจากหนังสือเล่มแรก แล้วมาอ่านอีกหนึ่งบรรทัดของหนังสือเล่มที่สอง ทำเช่นนี้สลับกันไปเรื่อย ๆ เชื่อได้ว่าท่านคงแทบจับใจความของหนังสือทั้งสองเล่มไม่ได้ เนื่องจากคนส่วนใหญ่มีความคุ้นเคยกับการคิดการทำงานแบบอนุกรม ดังนั้นจึงยากที่เราจะเปลี่ยนนิสัยมาคิดหรือทำอะไรแบบขนาน เช่นเดียวกันกับการเขียนโปรแกรมจึงไม่ง่ายนักที่จะเปลี่ยนโปรแกรมที่คำนวณแบบอนุกรมมาเป็นแบบขนาน และที่ยากไปกว่านั้นหลายสิบเท่าคือการที่ผิดพลาดในโปรแกรมที่คำนวณแบบขนาน ด้วยเหตุนี้ โปรแกรมเมอร์ที่เขียนโปรแกรมที่คำนวณแบบขนานบนเครื่องคอมพิวเตอร์แบบหลายโปรเซสเซอร์ จึงต้องการเครื่องมือที่จะช่วยให้การเขียนโปรแกรมเป็นไปอย่างมีประสิทธิภาพและทำงานได้ถูกต้องตามที่ต้องการ เครื่องมือเหล่านี้อาจจะมาในรูปของ subroutine library หรือการขยายความสามารถของภาษาคอมพิวเตอร์ที่มีอยู่

ในปัจจุบันเครื่องคอมพิวเตอร์แบบหลายโปรเซสเซอร์ได้รับการออกแบบและสร้างขึ้นมาหลายแบบ เราสามารถแบ่งออกได้เป็น 2 กลุ่มคือ กลุ่มคำสั่งเดี่ยวหลายข้อมูล (SIMD : Single-Instruction Multiple-Data) และกลุ่มหลายคำสั่งหลายข้อมูล (MIMD : Multiple-Instruction Multiple-Data) หลักการทำงานง่าย ๆ ของกลุ่ม SIMD คือ คอมพิวเตอร์ในกลุ่มนี้จะมีหน่วยคำนวณ (ALU) มากกว่าหนึ่ง แต่มีโปรแกรมอยู่ชุดเดียวที่คอยควบคุมการทำงานของ



หน่วยคำนวณเหล่านี้ ในแต่ละคำสั่งของโปรแกรม หน่วยคำนวณหลาย ๆ หน่วยจะคำนวณอย่างอิสระต่อกันจากข้อมูลคนละตัว ตัวอย่างคอมพิวเตอร์ในกลุ่มนี้ได้แก่ Illiac IV. ICL DAP MPP และ Connection Machine ส่วนหลักการทำงานของกลุ่ม MIMD คือ คอมพิวเตอร์ในกลุ่มนี้จะมีจำนวนโปรเซสเซอร์มากกว่าหนึ่ง แต่ละโปรเซสเซอร์สามารถคำนวณโปรแกรมได้อย่างเป็นอิสระซึ่งกันและกัน ตัวอย่างคอมพิวเตอร์ในกลุ่มนี้ได้แก่ ENCORE FLEX/32 iPSC และ RP3 เป็นต้น

ในบทความชุดนี้จะเน้นด้านการออกแบบโปรแกรมที่คำนวณแบบขนานสำหรับเครื่องคอมพิวเตอร์แบบหลายโปรเซสเซอร์ในกลุ่ม MIMD เพราะเครื่องในกลุ่มนี้ได้รับการออกแบบมาสำหรับการคำนวณทั่วไป ซึ่งต่างกับเครื่องในกลุ่ม SIMD ที่ได้รับการออกแบบมาสำหรับการคำนวณเฉพาะอย่าง

การแบ่งประเภทของคอมพิวเตอร์แบบ MIMD

ในการเขียนโปรแกรมแบบอนุกรม โปรแกรมเมอร์ไม่จำเป็นต้องรู้รายละเอียดด้านฮาร์ดแวร์ของคอมพิวเตอร์ที่ใช้งานอยู่ นอกจากว่าโปรแกรมเมอร์ต้องการให้โปรแกรมทำงานได้อย่างมีประสิทธิภาพสูงขึ้น ทำให้จำเป็นต้องรู้รายละเอียดบางอย่างของฮาร์ดแวร์ ในทางตรงกันข้าม สำหรับการเขียนโปรแกรมเพื่อการคำนวณแบบขนาน โปรแกรมเมอร์จำเป็นต้องเรียนรู้โครงสร้างของระบบคอมพิวเตอร์ที่ใช้งานอยู่พอสมควร เพราะระบบคอมพิวเตอร์แบบหลายโปรเซสเซอร์นั้นมีโครงสร้างแบบการคำนวณอัลกอริทึมบางอัน อาจคำนวณได้ดีในบางโครงสร้าง แต่อาจจะใช้งานไม่ได้เลยในโครงสร้างที่ต่างออกไป นอกจากนี้สไตล์การเขียนโปรแกรมยังขึ้นอยู่กับประเภทของคอมพิวเตอร์ที่ใช้งานอยู่ด้วย

คาร์ฟ (Karp) ได้แบ่งประเภทของคอมพิวเตอร์แบบหลายโปรเซสเซอร์ในกลุ่ม MIMD ออกเป็น 3 ประเภท ดังต่อไปนี้

1. ระบบหน่วยความจำร่วม (Shared-memory)

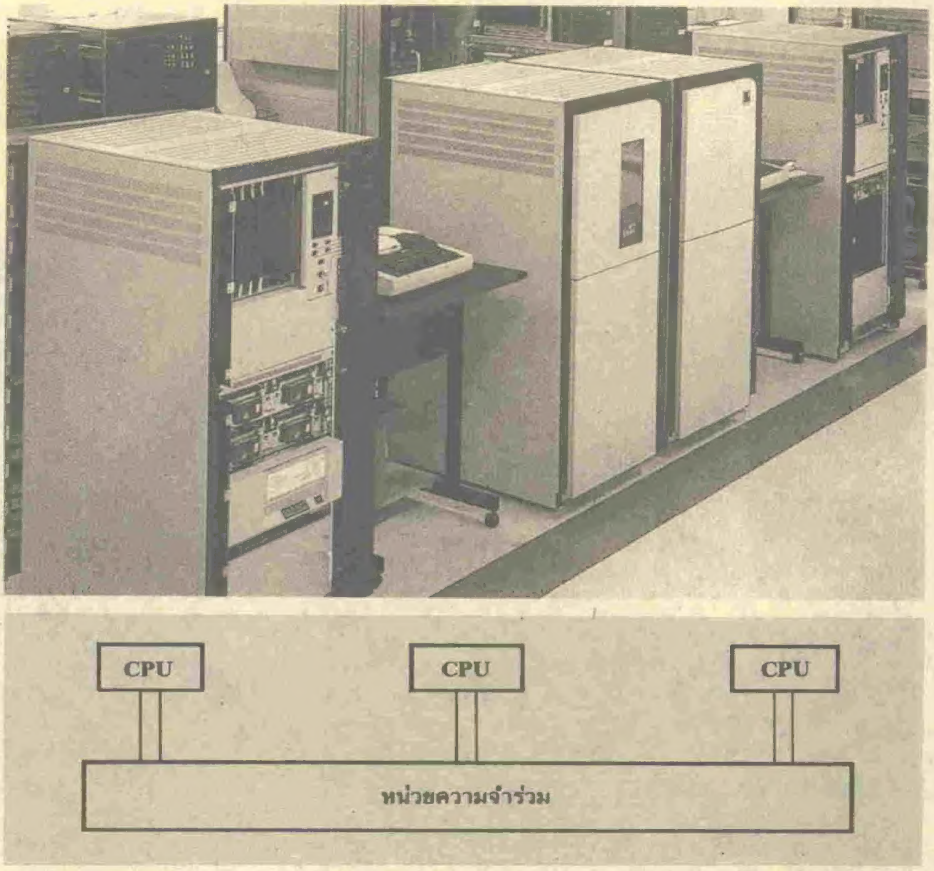
คอมพิวเตอร์ในระบบหน่วยความจำร่วมนี้ จะมีหน่วยความจำกองกลางที่ทุกโปรเซสเซอร์สามารถใช้งานได้ ดังแสดงในรูปที่ 2

โปรเซสเซอร์แต่ละหน่วยอาจจะมีหน่วยความจำย่อยที่เป็นของตัวเอง เช่น รีจิสเตอร์ หรือ แคช (cache) โดยผู้ใช้งานไม่ต้องคอยกังวลว่าข้อมูลที่อยู่บนหน่วยความจำย่อยส่วนตัวเหล่านี้จะต่างไปจากข้อมูลเดียวกันในหน่วยความจำส่วนกลาง เนื่องจากระบบควบคุม (operating system) จะคอยทำหน้าที่ควบคุมให้ข้อมูลที่ใช้งานเป็นข้อมูลที่ต้องการอยู่ตลอดเวลา

จุดเด่นของระบบหน่วยความจำร่วมคือ เวลาที่ใช้ในการอ่าน/เขียนข้อมูลระหว่างหน่วยความจำกับโปรเซสเซอร์ มีค่าคงที่ไม่ขึ้นอยู่ว่าเป็นโปรเซสเซอร์ตัวไหน จุดอ่อนของระบบนี้คือ หน่วยความจำร่วมไม่สามารถบริการโปรเซสเซอร์จำนวนมากได้ เพราะถ้ามีโปรเซสเซอร์จำนวนมากเกินไปจะทำให้หน่วยความจำร่วมไม่สามารถสนองความต้องการใช้ข้อมูลในหน่วยความจำได้เร็วเพียงพอ

2. ระบบส่งผ่านข้อมูล (Message passing)

ในระบบส่งผ่านข้อมูล โปรเซสเซอร์แต่ละตัวจะมีหน่วยความจำเป็นของ



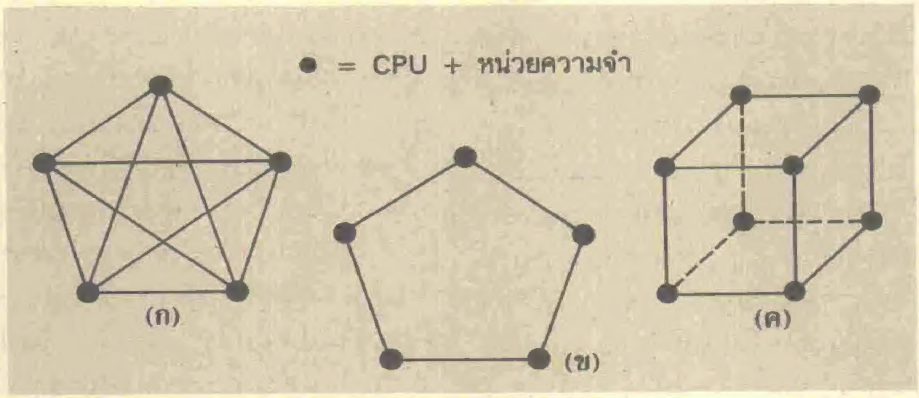
รูปที่ 2 คอมพิวเตอร์แบบหลายโปรเซสเซอร์ ระบบหน่วยความจำร่วม

ตัวเองโดยไม่มีหน่วยความจำที่ใช้ร่วมกันระหว่างโปรเซสเซอร์เลย วิธีเดียวที่จะให้โปรแกรมซึ่งคำนวณอยู่บนโปรเซสเซอร์แต่ละตัวสามารถใช้ข้อมูลร่วมกันได้นั้น โปรแกรมเมอร์จะต้องเขียนคำสั่งให้มีการส่งข้อมูลจากโปรเซสเซอร์หนึ่งไปยังอีกโปรเซสเซอร์หนึ่ง เวลาที่ใช้ในการอ่าน/เขียนข้อมูลระหว่างโปรเซสเซอร์กับหน่วยความจำมีค่าเปลี่ยนแปลงไปตามระยะทางระหว่างโปรเซสเซอร์ที่ต้องการข้อมูล และโปรเซสเซอร์ที่เป็นเจ้าของหน่วยความจำที่มีข้อมูลนั้นอยู่ ดังนั้นประสิทธิภาพในการทำงานของอัลกอริทึมจึงขึ้นอยู่กับการวางแผนให้ข้อมูลอยู่ที่หน่วยความจำที่เหมาะสม เพื่อลดระยะเวลาในการส่งผ่านข้อมูลระหว่างโปรเซสเซอร์ให้มีค่าต่ำที่สุด

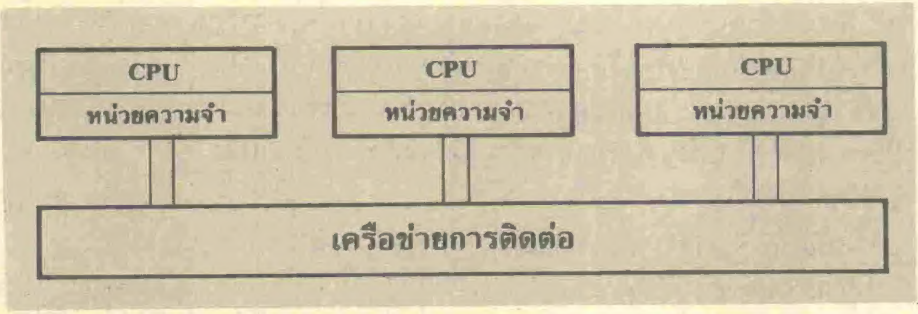
วิธีเชื่อมต่อระหว่างโปรเซสเซอร์ในระบบส่งผ่านข้อมูลนั้นมีอยู่หลายวิธี วิธีที่ง่ายที่สุดคือ การเชื่อมต่อแบบสมบูรณ์ดังในรูปที่ 3 (ก) โปรเซสเซอร์แต่ละตัวจะมีสายต่อไปยังโปรเซสเซอร์ทุกตัวที่เหลือ แต่วิธีนี้ไม่เหมาะกับระบบที่มีจำนวนโปรเซสเซอร์มาก ๆ เพราะจำนวนสายเชื่อมต่อจะเพิ่มขึ้นอย่างทวีคูณ จนเกือบจะเป็นไปไม่ได้ในแง่ปฏิบัติ เช่นระบบที่ประกอบด้วยหนึ่งแสนโปรเซสเซอร์ ถ้าเราใช้การเชื่อมต่อแบบสมบูรณ์ จำนวนการเชื่อมต่อทั้งหมดจะมีประมาณ 50 ล้านจุด เพราะฉะนั้นผู้ออกแบบจึงจำเป็นต้องหลีกเลี่ยงไปใช้วิธีการเชื่อมต่อที่ลดจำนวนสายลง แต่ผลที่ตามมาคือการส่งผ่านข้อมูลทำได้ช้าลง ตัวอย่างการเชื่อมต่อที่ได้รับความนิยมคือการเชื่อมต่อแบบวงแหวน และแบบลูกบาศก์ ดังที่แสดงในรูปที่ 3 (ข) และ 3 (ค) ตามลำดับ

จากรูปที่ 3 เราจะเห็นได้ว่าในวิธีการเชื่อมต่อแบบสมบูรณ์ โปรเซสเซอร์หนึ่งสามารถส่งข้อมูลไปยังอีกโปรเซสเซอร์หนึ่งได้โดยตรง ต่างกับวิธีการเชื่อมต่อแบบอื่นที่จำเป็นจะต้องส่งข้อมูลผ่านหลาย ๆ โปรเซสเซอร์ก่อนที่จะถึงผู้รับได้ ความแตกต่างระหว่างวิธีการเชื่อมต่อเหล่านี้ มีผลกระทบต่อการออกแบบอัลกอริทึมของโปรแกรมอย่างมาก เพราะอัลกอริทึมบางอย่างที่ทำงานได้อย่างมีประสิทธิภาพลดลงในเครื่องที่มีวิธีการเชื่อมต่ออีกแบบหนึ่ง

จุดเด่นของระบบส่งผ่านข้อมูลคือ สามารถเชื่อมต่อโปรเซสเซอร์เป็นจำนวนมากได้ จำนวนโปรเซสเซอร์ที่มากที่สุดที่มีในปัจจุบันคือ 65536 ตัว



รูปที่ 3 การเชื่อมต่อของคอมพิวเตอร์แบบหลายโปรเซสเซอร์ระบบส่งผ่านข้อมูลแบบต่าง ๆ (ก) แบบสมบูรณ์ (ข) แบบวงแหวน (ค) แบบลูกบาศก์



รูปที่ 4 คอมพิวเตอร์แบบหลายโปรเซสเซอร์ระบบผสม

3. ระบบผสม (Hybrid)

เครื่องคอมพิวเตอร์ในระบบผสมจะมีคุณสมบัติของทั้งระบบหน่วยความจำร่วมและระบบส่งผ่านข้อมูลรวมกัน โปรเซสเซอร์แต่ละตัวจะมีหน่วยความจำเป็นของตัวเอง ดังแสดงในรูปที่ 4 แต่ว่าระบบควบคุม (operating system) จะทำให้ผู้ใช้โปรแกรมเมอร์ มองเห็นเสมือนหนึ่งว่าหน่วยความจำส่วนตัวเหล่านั้นเป็นของส่วนกลาง ข้อมูลที่ถูกเรียกใช้จะถูกส่งผ่านเครือข่ายการติดต่อไปยังโปรเซสเซอร์ที่เรียกใช้โดยระบบควบคุมอย่างอัตโนมัติ

การเขียนโปรแกรมบนเครื่องคอมพิวเตอร์ระบบนี้ จะเหมือนกับระบบหน่วยความจำร่วม เพราะว่าผู้ใช้มองเห็นหน่วยความจำทั้งหมดเป็นของกลาง แต่ประสิทธิภาพของโปรแกรมจะคล้ายกับของระบบส่งผ่านข้อมูล เนื่องจากการกระจายข้อมูลตามโปรเซสเซอร์ต่าง ๆ มีผลกระทบต่อเวลาในการเรียกใช้ข้อมูลนั้น ๆ

ในฉบับต่อไป (ตอนที่ 2) ผู้เขียนจะกล่าวถึงปัญหาในการเขียนโปรแกรมเพื่อการคำนวณแบบขนานในระบบต่าง ๆ และแนะนำเครื่องมือช่วยที่จำเป็นในการเขียนโปรแกรม